

REMARKS

The Examiner continues to maintain the rejections of claims 1-13 under 35 U.S.C. 102(b) or 35 U.S.C. 103(a) as being anticipated by or being unpatentable over Swift et al ("Swift"). To the extent that PDUs (Packet Data Units) and ASPs are well known in the communication protocol arts, Applicants acknowledge such as prior art. However Applicants do not necessarily agree with the Examiner's conclusion regarding combination with Swift to produce Applicants' claimed invention. Further the Examiner states that Swift allows a user to select one or more network protocols from a selection, referencing Figs. 1 and 8 and page 16, paragraph 3 through page 17 paragraph 1. Applicant continues to traverse the Examiner's conclusions about the teachings of Swift.

In contradistinction to Applicants' claimed invention Swift discloses a controlled test environment 100 for verifying the response of target objects 114 to the reception of message sequences 106, which environment includes a message sequence generator (MSG) 102 for defining and generating messages that correspond to actual message sequences transmitted by network source objects (devices capable of communicating messages on a network) to the target objects in a production network (as opposed to a controlled network) in order to simulate the production network. The MSG is connected to a TCP/IP network 103, or any other network that can transfer the messages, to which also is connected a data collector 108. One or more store forward files 110 are accessible by the data collector, one or more data distributors 112 monitor the store forward files and the target objects receive the message sequence from the appropriate data distributor. The data distributors convert the message sequence to the appropriate

protocol for the target object. The ability of the data distributors to communicate via a wide variety of protocols enables a single MSG, which is capable only of a single protocol, to send messages to numerous target objects regardless of the protocol used by the target object. Swift selects a target object to receive messages, generates the message sequence and transmits the message sequence to the object. In more detail, when a user wishes to test a target system a message sequence engine 218 displays a message generation window 404 that has a message sequence text entry field 406. To establish a message sequence the user may either search a message database 214 for a previously saved sequence definition or manually add a sequence item by specifying the network source object 304 (source being simulated) and a data message 306. Once the message sequence has been specified by a series of sequence items, the user selects the target object to which the message sequence is to be sent. A registration message 104 is sent to the data collector to establish a session between the message sequence engine and the data collector and to identify the target object. There is no mention in Swift of emulating any particular protocol layer – Swift merely sends messages to the target object and observes the response of the target object. Swift has no control over any protocol selection except to the extent of specifying the target object – any protocol redefinition occurs at the data distributor, i.e., the MSG is oblivious to the protocol of the target object.

Since Swift does not emulate any particular protocol layer, there is no need for Swift to specify abstract communication interfaces for an emulated protocol layer. The Examiner's statement that the fact that the MSG transmits messages over a TCP/IP network equates to "selecting a protocol layer to be emulated" as recited by Applicants

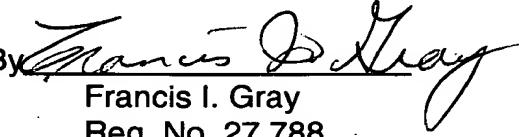
in the independent claims 1 and 8 is clearly erroneous. Further Swift is not setting up a communication procedure, but is generating messages for transmission to a target object. There is absolutely no similarity between the two except that messages may be sent to the instance under test as part of the communication between the protocol tester and the instance under test when the communication procedure is executed.

Determining the message sequence is not the same as setting up a communication procedure, i.e., the message sequence (if required) is determined after the communication protocol is set up. The claimed invention deals with setting up the procedure, not with transmitting messages. Transmitting messages does not require any knowledge of abstract communication interfaces between protocol layers. Since there is nothing in Swift that teaches or suggests how to set up a communication procedure, claims 1-13 are deemed to be allowable as being neither anticipated nor rendered obvious to one of ordinary skill in the art by Swift.

In view of the foregoing argument allowance of claims 1-13 is urged, and such action and the issuance of this case are requested.

Respectfully submitted,

JOERG EHRHARDT et al

By 
Francis I. Gray
Reg. No. 27,788
Attorney for Applicant

TEKTRONIX, INC.
P. O. Box 500, MS 50-LAW
Beaverton, Oregon 97077
(503) 627-7261

7057 US